Collision Avoidance and Resolution Multiple Access with Transmission Groups

Rodrigo Garcés and J.J. Garcia-Luna-Aceves
Baskin Center for Computer Engineering and Information Sciences
University of California at Santa Cruz
Santa Cruz, CA 95064, USA
garces@cse.ucsc.edu jj@cse.ucsc.edu

Abstract

The CARMA-NTG protocol is presented and analyzed. CARMA-NTG dynamically divides the channel into cycles of variable length; each cycle consists of a contention period and a group-transmission period. During the contention period, a station with one or more packets to send competes for the right to be added to the group of stations allowed to transmit data without collisions; this is done using a collision resolution splitting algorithm based on a request-to-send/clear-to-send (RTS/CTS) message exchange with non-persistent carrier sensing. CARMA-NTG ensures that one station is added to the group transmission period if one or more stations send requests to be added in the previous contention period. The group-transmission period is a variable-length train of packets, which are transmitted by stations that have been added to the group by successfully completing an RTS/CTS message exchange in previous contention periods. As long as a station maintains its position in the group, it is able to transmit data packets without collision. An upper bound is derived for the average costs of obtaining the first success in the splitting algorithm. This bound is then applied to the computation of the average channel utilization in a fully connected network with a large number of stations. These results indicate that collision resolution is a powerful mechanism in combination with floor acquisition and group allocation multiple access.

1 Introduction

Several medium access control (MAC) protocols have been proposed over the past few years that are based on three- or four-way handshake procedures meant to reduce the number of collisions among data packets, thereby providing better performance than the basic ALOHA or CSMA protocols [1], [2], [3], [4], [5], [7], [11], and [14]. The concept of *Group Allocation Multiple Access* (GAMA) was first introduced by Muir and Garcia-Luna-Aceves [12] to provide performance guarantees in asynchronous MAC protocols. A GAMA protocol dynamically divides the channel into cycles of variable length; each cycle consists of a contention period and a group-transmission period. During the contention period, a station with a message to send competes for the right to be added to the transmission group; this is done using a request-to-send/clear-to-send (RTS/CTS) message exchange with carrier sensing. The group-transmission period is a variable-length train of packets, which are transmitted by stations that have been added to the transmission group by successfully competing for it. As long as a station maintains its position in a transmission group, it transmits packets without collisions. Variants of this basic strategy can be designed using different types of contention-based MAC protocols like ALOHA or CSMA to transmit RTS packets into the channel.

We have recently shown [6] that collision resolution when applied to the RTS/CTS handshake used in many MAC protocols can improve the throughput of the system substantially. In this paper we describe and analyze a new channel access protocol that combines group allocation multiple access with collision resolution. We call the resulting protocol *Collision Avoidance and Resolution Multiple Access* protocol with non-persistent trees and transmission groups (CARMA-NTG). CARMA-NTG provides dynamic reservations of the channel, together with collision resolution of the reservations requests based on a tree-splitting algorithm [8]. Like GAMA [13], CARMA-NTG builds a dynamically-sized cycle that grows and shrinks depending upon traffic demand. Each cycle consists of a contention period and a group-transmission period during which one or more stations transmit data packets without collision. A position in the transmission group is allocated to an individual station during the contention period, and a station can continue to transmit in this position as long as it has data to send. Stations compete to acquire the right to be in the transmission group based on a tree-splitting algorithm.

CARMA-NTG is more attractive than previous dynamic reservation schemes for wireless nets [8], [9], [15]–[17] in that it does not require time synchronization and in that it does not require the definition of control frames of fixed duration over which the slots for the data frame can be reserved. It is also more attractive than token passing schemes in that no fixed schedule exists for passing the token and no special case needs to be taken of the possibilities of losing the token (the group allocation process simply restarts).

Section 2 describes CARMA-NTG, which uses non-persistent carrier sensing for the transmission of RTSs and a tree-splitting algorithm that resolves collisions by allowing a single station to succeed and be added to the transmission group. Section 3 computes an upper bound on the times associated with the eventual first successful allocation of a station to the group-transmission period involved in a collision resolution tree. The importance of these bounds is that they are independent of the number of stations in the network. Section 4 uses the bounds to compute the average throughput achieved by CARMA-NTG when a very large population of nodes is assumed. As the average duration of transmission groups or the persistence of the station in the groups increases, CARMA-NTG becomes in effect TDMA, giving every station in the transmission group a slot in each cycle. Section 5 offers our concluding remarks.

[0] This work was supported in part by DARPA under Grants DAAB07-95-C-D157 and DAAH04-96-10210

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1997**	2. REPORT TYPE	3. DATES COVERED **00-00-1997 to 00-00-1997**

4. TITLE AND SUBTITLE
Collision Avoidance and Resolution Multiple Access with Transmission Groups

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **9**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

2 CARMA-NTG

CARMA-NTG maintains a dynamically-sized data interval and a dynamically-sized contention interval based on RTS collision resolution, resulting in a stable network even when the load on the channel is high.

In CARMA-NTG, each cycle is composed of a contention period and a group-transmission period. The group-transmission period consists of one or more stations transmitting data packets without collision. Each station in the transmission group is allowed to transmit one data packet per cycle. The number of transmission periods k in a group-transmission period ranges from 0 to r, where $\frac{1}{r}$ is the minimum guaranteed transmission rate in the network. Each station in the network knows the number of members in the transmission group, its own position within the group, and the beginning of each group-transmission period. Each station in the system is aware of every successful RTS/CTS exchange. During the group-transmission period, each station allocated to the group transmits its packet as soon as the packet from the previous transmitting station is received. The maximum spacing between packets is twice the propagation delay. If a packet from the preceding station is not received within this period, then the station is assumed to have failed and its transmission period is removed from the transmission group; a new contention period starts after that.

During the contention period, the stations in the network reserve a space in the group-transmission period. To understand better how the contention protocol works in CARMA-NTG, consider how RTS collisions are solved in GAMA [12].

GAMA requires a station that wishes to send one or more packets to acquire the right to be part of the transmission group before transmitting the data packets. Stations follow a non-persistent CSMA strategy for the transmission of RTSs with which they request to be added to the transmission group. The sender of an RTS listens to the channel for one maximum round-trip time plus the time needed for the destination to send a CTS. If the CTS is not corrupted and is received within the time limit, the station is added to the transmission group. Although each station transmits an RTS only when it determines that the channel is free, during a predefined amount of time in the contention period, a collision with other RTSs may still occur due to propagation delays. RTSs are vulnerable to collisions for time periods equal to the propagation delays between senders of RTSs. During these periods, multiple stations may sense the channel free and also send RTSs, thus causing collisions.

As specified in [13], GAMA protocols solve collisions by backing off and rescheduling RTS transmissions. As with CSMA protocols, this procedure yields good results if the RTS traffic is low; however, the probability of RTS collisions increases as the rate of RTS transmissions increases, with a corresponding decrease of system throughput. Eventually, as the RTS transmission rate increases, the constant RTS collisions cause the channel to collapse, bringing the flow of data packets to a halt when no new transmission groups can be started.

CARMA-NTG uses carrier sensing for the transmission of RTSs and a tree-splitting algorithm to resolve collisions of RTSs. Each station must know the maximum number of stations allowed in the system and the maximum propagation delay in the network. For simplicity, our analysis assumes that time is slotted, with a slot lasting one maximum propagation delay.

2.1 Contention Period Description

During the contention period, a station with one or more packets to send competes for the right to be added to the group of stations allowed to transmit data without collisions; this is done using a collision-resolution splitting algorithm based on a request-to-send/clear-to-send (RTS/CTS) exchange with non-persistent carrier sensing. The splitting algorithm in CARMA-NTG terminates the contention period as soon as the first successful RTS/CTS exchange takes place. The contention period allows only one single station to succeed in being added to the transmission group per cycle. In this paper, we assume that stations are allowed to contend to be added to the transmission group if they have packets to send within one maximum propagation delay from the end of a group-transmission period.

Each station is assigned a unique identifier, a stack and two variables ($LowID$ and $HiID$). $LowID$ is initially set to 1 and denotes the lowest ID number that is allowed to send an RTS. $HiID$ is the highest ID number that is allowed to send an RTS. Together they constitute the allowable ID-number interval that can send RTSs. If the ID of the station is not within this interval or the station has already been assigned to the transmission group, it cannot send its RTS. As we describe subsequently, the stack is simply a storage mechanism for ID-intervals that are waiting for permission to send an RTS. Based on these variables, a station can be in one of the following five states:

PASSIVE the station has no local packets pending and has not detected the start of a transmission period

BACKOFF the station has local packets pending and could not be added to the transmission group

REMOTE the station has no local packets and knows that there is a transmission group

RTS the station is trying to acquire a position in the transmission group

XMIT the station is part of the transmission group

When a PASSIVE station obtains one or multiple packets to send, it first listens to the channel. If the channel is clear (i.e., no carrier is detected), the station enters the RTS state by sending an RTS. The sender then waits and listens to the channel for one maximum round-trip time plus the time needed for the destination to send a CTS. When the originator receives the CTS from the destination, it is added to the transmission group and enters the XMIT state, which starts following the CTS.

On the other hand, if a PASSIVE station obtains packets to send but detects carrier, it enters the BACKOFF state. The station then computes a random backoff time and attempts to enter the transmission group after that time.

If the CTS is corrupted or is not received within the time limit, the sender of the RTS, as well as all other stations in the system, know that a collision has occurred. As soon as the first collision takes place, every station divides the ID-interval $(LowID, HiID)$ into two ID-intervals. The first ID-interval, which we will call the backoff interval, is $(LowID, \lceil \frac{HiID-LowId+1}{2} \rceil)$, while the second ID-interval, the allowed interval, is $(\lceil \frac{HiID-LowId+1}{2} \rceil + 1, HiID)$. Each station in the system updates the stack by executing a PUSH stack command, where the key being pushed is the backoff interval. After this is done, the station updates $LowID$ and $HiID$ with the values from the allowed interval. This procedure is repeated each time a collision is detected.

If a station is in REMOTE state and obtains one or more packets to send, it senses the channel for one maximum round-trip delay, which is the largest gap between data packets in a group-transmission period. If the station detects no carrier, it sends its RTS; otherwise, it goes to the BACKOFF state.

Only those stations in the RTS state at the time the collision occurred can persist trying to be added to the transmission group. The persistence continues until one station is added to the transmission group. The stations in the transmission group do not send an RTS until they leave the group. All other stations must attempt to join the transmission group after a random backoff delay.

All stations in the allowed ID-interval that are in the RTS state try to re-transmit an RTS. If none of the stations within this interval request the channel, the channel will be idle for a time period equal to a maximum channel delay (τ). At this point, an update of the stack and the variables $LowID$ and $HiID$ is made. Each station executes a POP command in the stack. This new ID-interval now becomes the new $HiID$ and $LowID$. The same procedure takes place if, during the first collision, only one station is requesting the channel. The originator receives the CTS from the destination and is added to the transmission group, after which the sender of the RTS transitions to the XMIT state. The total time for a successful addition to the transmission group is equal to the duration of an RTS, a CTS, plus two channel delays. If two or more stations request the channel then, a second collision occurs. The stations in the allowed ID-interval are again split into two new ID-intervals; the stack and the variables for each station are updated. The duration of this event is equal to the collision time plus the channel delay. The algorithm repeats these steps until one station is added to the transmission group. A successful RTS/CTS exchange indicates the termination of the tree-splitting algorithm and the beginning of the next group-transmission period. To ensure fairness within the splitting algorithm, the position of the stations in the tree (which is equivalent to changing the ID number) can vary after each contention period. For example, the ID numbers of the stations can rotate cyclically. Each station increases its ID number by one and the last station takes the ID number of the previous first station.

The collision resolution algorithm is only used during the contention period and its sole purpose is to assign one new station to the transmission group. In each step of the contention period, one of the following three cases can occur:

Case 1–Idle: There are no RTS packets in any of the stations in subtree T_r; therefore, the channel is idle and no new member is assigned to the transmission group. It lasts τ seconds.

Case 2–Success: There is only one RTS packet in the subtree T_r; therefore, there is no collision and a station acquires the floor and is assigned to the transmission group. It lasts $2\gamma + 2\tau$ seconds.

Case 3–Collision: There are two or more stations (leaves) in the subtree T_r sending an RTS; therefore a collision occurs. It lasts $\gamma + \tau$ seconds.

2.2 Example

To illustrate the CARMA-NTG protocol let us use as an example a system with four stations. Assume that, at time t_0, station n_{11} is the only station assigned to the group-transmission period with five data packets to transmit and the last cycle has ended and we are at the beginning of a new cycle, that is, in the contention period. The stations compete to be added to the transmission group. At time t_0, Case 3 occurs with station n_{00} and n_{01} each sending an RTS in the same slot, while station n_{10} and station n_{11} do not request to be added to the transmission group (see Fig. 1). Station n_{11} does not need to send a request since it is already assigned to the transmission group and all other stations knows that it still has five data packets to transmit. Let station n_{01} have three data packets to send while station n_{00} has only one data packet to transmit. At time t_0 the first collision occurs, all stations in the system notice the beginning of the resolution algorithm, as well as the beginning of the contention period. All stations update their stacks and their $LowID$ as well as their $HiID$ values. After $\gamma + \tau$ seconds have elapsed, stations n_{00} and n_{01} backoff and will therefore wait until the collisions in the allowed-ID set are resolved. They both are excluded from sending RTSs. Stations n_{10} and n_{11} are allowed to request the channel. Since stations n_{11} and n_{10} in tree T_1 do not wish to be added to the transmission group, an idle contention period occurs (Case 1). After τ seconds, all stations notice that the channel is idle, which means that there were no collisions in tree T_1. All the stations in the system must update their intervals and the stack. They execute a POP-stack command and the new allowable interval is $(00, 01)$; therefore, T_0 can proceed to solve its RTS collisions. Both stations n_{00} and n_{01} transmit an RTS control packet and Case 3 occurs again. Because a collision occurred, after $\gamma + \tau$ seconds the interval is split, that is, the subtree T_0 is split into two halves, T_{00} and T_{01}. Station n_{01} is within the allowable interval while the n_{00} station must wait; its interval is the top of the stack. Since T_{01} has only one station requesting the channel, station n_{01} is assigned to the transmission group after $2\gamma + 2\tau$ seconds. During the successful RTS/CTS exchange, all other stations know that station n_{01} has three packets to send. The contention period ends and station n_{01} begins the transmission of its data packet immediately after it has read the packet transmitted by the preceding group member, which is station n_{11}. After $2\delta + 2\tau$ seconds, station n_{11} has four data packets left while station n_{01} has two packets. Let us assume that no new data packets arrive. In the next contention period, since stations n_{11} and n_{01} are already members of the transmission group, stations n_{00} and n_{10} can compete to be added to the transmission group. If n_{00} is the only station requesting to be added to the transmission group, it is added, marking the end of the second contention period. The length of the group-transmission period is $3\delta + 3\tau$ seconds. At this point, station n_{00} does not have any more packets to send and is removed from the transmission group, while stations n_{11} and n_{01} remain in the transmission group. Station n_{11} still has three data packets to send while station n_{01} needs one more run. The size of the next contention period is only τ seconds; because no new RTSs requests are sent. The duration of the next group-transmission period is $2\delta + 2\tau$ seconds, after which station n_{01} is removed from the transmission group. The next two contention periods are of size τ seconds each, while the group-transmission

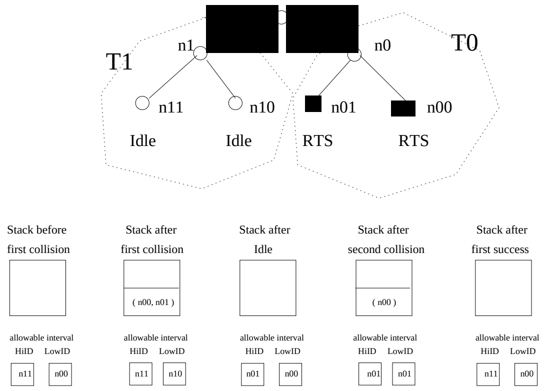

Figure 1: Tree structure for an example with $n = 4$ stations and $m = 2$ stations are requesting to be admitted into the transmission group.

periods are of size $\delta + \tau$ seconds. Finally, station n_{11} is removed and the transmission group is empty.

3 Average Collision Resolution Costs

In this section, we present upper bounds for the average costs of resolving RTS collisions. Every station can listen to the transmissions of any other station. For the purpose of our analysis, we assume that (a) the channel introduces no errors, so packet collisions are the only source of errors, and stations detect such collisions perfectly, (b) two or more transmissions that overlap in time in the channel must all be re-transmitted, (c) a packet propagates to all stations in exactly τ seconds [10], (d) time is slotted in τ-second slots, and (e) no failure of stations occur. The average size of a data packet is δ seconds, and RTS and CTS packets are of size γ seconds. Both δ and γ are assumed to be multiples of τ.

3.1 Average Cost Analysis

First, we present a combinatorial calculation of the average collision resolution costs for a system with n stations and m RTSs arriving during the time period τ. Because each station is assigned one or zero RTS at any given time, a leaf of the binary tree which corresponds to a station in the system is assigned an "RTS" or an "idle", depending on whether or not it has an RTS to send.

The binary tree is a structure defined on a finite set of nodes composed of three disjoint sets: a root node, a binary tree called a left subtree, and a binary tree called a right subtree. As we have described, there are only three possible cases to consider for the resolution of RTS collisions: idle, success, or collision. For each of these cases, we wish to find a recursive equation expressing its average cost, i.e., the number of subtrees starting from the root that need to be visited before the first successful RTS/CTS exchange. To do so, we assign three distinct average cost values: $\overline{\mathcal{Z}}(n,m)$ for the idle case, $\overline{\mathcal{S}}(n,m)$ for the success case, and $\overline{\mathcal{C}}(n,m)$ for the collision case. These three costs depend on the number of leaves n and the number of stations with RTSs, m. They represent an average number over all the possible permutations of m RTSs in n total stations. What each of these costs actually means and what rules apply to each of the three cases can best be explained by means of a simple example.

The number of permutations of a tree with four leaves ($n = 4$), given two out of the four stations are requesting the channel simultaneously ($m = 2$), is six. In our previous example, stations n_{00} and n_{01} each sends an RTS in the same time slot, while station n_{10} and station n_{11} remain idle. This tree represents just one of the six possible permutations. Let us assign the index i to our tree and calculate $\mathcal{Z}_i(4,2), \mathcal{S}_i(4,2)$ and $\mathcal{C}_i(4,2)$ respectively. Starting with the root node, the first thing that happens is a collision because two stations are competing for the channel. After updating the stack and the intervals the next step is to take subtree T_1. Since no RTSs are present, the total zero cost is $\mathcal{Z}_i = 1$. Following the algorithm rules, we take subtree T_0 and again a collision occurs. The total collision cost is $\mathcal{C}_i = 2$. The next step is to go to node n_{01}, which is a successful RTS/CTS exchange. At this point the collision resolution algorithm ends, since a success has been achieved. The total successful cost is $\mathcal{S}_i = 1$. What we have counted is the number of subtrees that have collisions ($\mathcal{C}_i = 2$), the number of subtrees that only have one RTS in them ($\mathcal{S}_i = 1$), as well as the number of subtrees that are idle ($\mathcal{Z}_i = 1$). The same counting procedure can be repeated for each of the $\binom{4}{2} = 6$ permutations of trees with $n = 4$ and $m = 2$. The six possible combinations contribute equally to the total average costs $\overline{\mathcal{C}}(4,2), \overline{\mathcal{S}}(4,2)$ and $\overline{\mathcal{Z}}(4,2)$. The average for each of the three types of costs can be calculated by adding each individual permutation cost and by dividing by the total number of permutations.

For counting purposes, a subtree that has no RTS stations or only one RTS station does not need to be explored further. Counting can stop there and add one unit either to $\mathcal{Z}$ or to $\mathcal{S}$. It is interesting to observe that $\mathcal{S}$ is always equal to 1. Based on the example we can deduce the general rules shown in Table 1.

Rule 1:	$\mathcal{C}(n,0) = 0$	**Rule 4:**	$\mathcal{S}(n,m) = 1$
Rule 2:	$\mathcal{C}(n,1) = 0$	**Rule 5:**	$\mathcal{Z}(n,0) = 1$
Rule 3:	$\mathcal{S}(n,0) = 0$	**Rule 6:**	$\mathcal{Z}(n,1) = 0$

Table 1: Average Cost Rules for splitting algorithm until the first success is reach

3.2 Average Collision Cost C(n,m)

In our example of the binary tree with four leaves ($n = 4$) and station n_{00} and n_{01} each sending an RTS while station n_{10} and station n_{11} remain idle, we found the collision cost to be $\mathcal{C}_i(4,2) = 2$ units. This tree can be visualized as two disjoint subtrees and the parent node. Similarly, the total collision cost can be expressed as the cost of the right subtree, plus the cost of the left subtree plus one unit cost for the root of the tree. This result can be extended to the general case, yielding the following equation:

$$\mathcal{C}_{root\ tree} = \mathcal{C}_{right\ subtree} + \mathcal{C}_{left\ subtree} + 1 \qquad (1)$$

In addition to our example, there are five other possible ways to distribute two stations with an RTS to send in four positions. Table 2 shows all six cases and the collision cost $\mathcal{C}_i$ associated with each of them.

n_{11}	n_{10}	n_{01}	n_{00}	**C**	**S**	**Z**
idle	idle	RTS	RTS	2	1	1
idle	RTS	idle	RTS	1	1	0
idle	RTS	RTS	idle	1	1	0
RTS	idle	idle	RTS	1	1	0
RTS	idle	RTS	idle	1	1	0
RTS	RTS	idle	idle	2	1	0

Table 2: Cost Table for all possible permutations for a tree with $n = 4$ and $m = 2$.

To calculate the average cost, we sum each individual cost and divide this result by six. The average cost $\overline{\mathcal{C}}(4,2)$ can be expressed in the following compact equation:

$$\overline{\mathcal{C}}(4,2) = 1 + \frac{\binom{2}{0}\binom{2}{2}}{\binom{4}{2}}\overline{\mathcal{C}}(2,2) + \sum_{i=2}^{2} \frac{\binom{2}{i}\binom{2}{2-i}}{\binom{4}{2}}\overline{\mathcal{C}}(2,i) = \frac{8}{6} \qquad (2)$$

The recursion splits the original $n = 4$ tree into two $n = 2$ subtrees plus the root node. This result can be extended to the average collision cost $\overline{\mathcal{C}}(n,m)$. Let $\alpha = \lceil n/2 \rceil$ be the number of leaves in the right subtree, while $\beta = n - \alpha$ is the number of leaves in the left subtree. If n is even, then $\alpha = \beta = \frac{n}{2}$, otherwise $\beta = \alpha - 1$. We need to consider three cases.

- Case 1: $m \leq \alpha$ and $m \leq \beta$.

$$\overline{\mathcal{C}}(n,m) = 1 + \frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}}\overline{\mathcal{C}}(\beta,m) + \sum_{i=0}^{m-1} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}}\overline{\mathcal{C}}(\alpha, m-i) \qquad (3)$$

- Case 2: $m \leq \alpha$ and $m > \beta$.

$$\overline{\mathcal{C}}(n,\, m) = 1 + \sum_{i=0}^{\beta} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \overline{\mathcal{C}}\,(\alpha,\, m\, -\, i) \tag{4}$$

- Case 3: $m > \alpha$ and $m > \beta$.

$$\overline{\mathcal{C}}(n,\, m) = 1 + \sum_{i=m-\alpha}^{\beta} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \overline{\mathcal{C}}\,(\alpha,\, m\, -\, i) \tag{5}$$

Each cost unit for $\overline{\mathcal{C}}(n,m)$ has a time overhead of $T_f = \gamma + \tau$ and contributes negatively to the overall throughput of the system.

The next average cost, $\overline{\mathcal{S}}(n,m)$, contributes positively to the overall throughput of the system and is very simple to calculate. Every contention period with one or more stations requesting the channel is guaranteed to have one successful RTS/CTS exchange, therefore, $\overline{\mathcal{S}}(n,m) = 1$. Each success cost unit has a time period of $2\gamma + 2\tau$. Each time an RTS is successful a station is added to the group-transmission period.

3.3 Average Idle Cost Z(n,m)

We now compute the average number of subtrees in a tree with n leaves and m stations that do not have any RTS to send. According to the rules in Table 1, no matter how big the tree or subtree is, if there is only one RTS leaf ($m = 1$), the idle cost will always be zero.

In our example of the tree with four leaves ($n = 4$) and two stations with an RTS to send ($m = 2$), we can plot a cost table with all the six permutation cases, as shown in Table 2. We stop counting as soon as the first RTS/CTS exchange has taken place.

To calculate the average cost for our example, we sum each of the individual costs and divide this result by six permutations. There is only one case in which $\overline{\mathcal{Z}} = 1$; all the rest are equal to 0. The total average idle case for this example can be expressed in the following compact equation:

$$\overline{\mathcal{Z}}(4,\, 2) = \frac{\binom{2}{0}\binom{2}{2}}{\binom{4}{2}} \left(1 + \overline{\mathcal{Z}}\,(2,\, 2)\right) + \frac{\binom{2}{2}\binom{2}{0}}{\binom{4}{2}} \overline{\mathcal{Z}}\,(2,\, 2) = \frac{1}{6} \tag{6}$$

This result can be extended to the average idle cost for any n leaves and m stations with an RTS to send, $\overline{\mathcal{Z}}(n,m)$. Similar to the previous section, we have again three distinct cases, yielding the following equations:

- Case 1: $m \leq \alpha$ and $m \leq \beta$.

$$\overline{\mathcal{Z}}(n,\, m) = \frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}} \left[1 + \overline{\mathcal{Z}}\,(\beta,\, m)\right] + \sum_{i=0}^{m-2} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \overline{\mathcal{Z}}\,(\alpha,\, m\, -\, i) \tag{7}$$

- Case 2: $m \leq \alpha$ and $m > \beta$.

$$\overline{\mathcal{Z}}(n,\, m) = \sum_{i=0}^{\beta} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \overline{\mathcal{Z}}\,(\alpha,\, m\, -\, i) \tag{8}$$

- Case 3: $m > \alpha$ and $m > \beta$.

$$\overline{\mathcal{Z}}(n,\, m) = \sum_{i=m-\alpha}^{\beta} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \overline{\mathcal{Z}}\,(\alpha,\, m\, -\, i) \tag{9}$$

The parameters α and β are the same as in Eqs. 3,4, and 5. The time period for each cost unit for the idle cost case is equal to the channel delay τ.

3.4 Upper Bound

Obviously, the upper bound of the average success cost $\overline{\mathcal{S}}(n,m)$ is equal to 1. We use mathematical induction to prove the upper bounds for the average idle cost $\overline{\mathcal{Z}}(n,m)$ and for the average collision cost $\overline{\mathcal{C}}(n,m)$. Let μ be define as the lower index and ν as the upper index of the summations in Eqs. 3,4, 5,7, 8, and 9. We know that there are three possible combinations, determining the indices of the summation. First if $m \leq \alpha$ and $m \leq \beta$, then $\mu = 0$ while the $\nu = m$. In the second case, $m \leq \alpha$ and $\beta > m$, then $\mu = 0$, while $\nu = \beta$. Finally, if m is greater than both α and β, both indices change to $\mu = m - \alpha$ and $\nu = \beta$. The parameter m cannot be at the same time $> \alpha$ and $\leq \beta$ since $\beta \leq \alpha$, therefore; we disregard this case.

Theorem 1: For all $m > 1$ and $n > 1$, $\overline{\mathcal{Z}}(n,m) \leq \frac{1}{2}$.

Proof: From Table 1, we know that $\overline{\mathcal{Z}}(n,0) = 1$, $\overline{\mathcal{Z}}(n,1) = 0$ and that $\overline{\mathcal{Z}}(n,n) = 0$. Let $n = 2$ and $m = 2$; therefore, $\overline{\mathcal{Z}}(2,2) = 0 \leq \frac{1}{2}$.

Now we assume that, for all $2 \leq n \leq \alpha$ and all $2 \leq m \leq \nu$, the conditions $\overline{\mathcal{Z}}(\alpha,m) \leq \frac{1}{2}$ are satisfied; we show that the condition holds for all $\overline{\mathcal{Z}}(n,m)$. There are three cases to consider.

Case 1: $m \leq \alpha$ and $m \leq \beta$: Then $\mu = 0$ while $\nu = m$; therefore, using Eq. 7 and substituting every $\overline{\mathcal{Z}}$ for $\frac{1}{2}$ we get

$$\overline{\mathcal{Z}}(n,\, m) \leq \frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}} + \frac{1}{2}\frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}} + \frac{1}{2}\sum_{i=0}^{m-2} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \tag{10}$$

For any binomial coefficient, $\binom{n}{k} = \frac{n-k+1}{k}\binom{n}{k-1}$. Hence,

$$\overline{\mathcal{Z}}(n,\, m) \leq \frac{1}{2} + \frac{2\beta\, -\, m\alpha\, +\, 2(1\, -\, m)}{2m} \frac{\binom{\beta}{m-1}}{\binom{n}{m}} \leq \frac{1}{2} \tag{11}$$

The proof for **Case 2** ($m \leq \alpha$ and $m > \beta$) and **Case 3** ($m > \alpha$ and $m > \beta$) can be done together starting with

$$\overline{\mathcal{Z}}(n,\, m) = \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \overline{\mathcal{Z}}\,(\alpha,\, m\, -\, i) \tag{12}$$

Therefore, we have

$$\overline{\mathcal{Z}}(n,\, m) \leq \sum_{i=\mu}^{\nu} \frac{1}{2} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \leq \frac{1}{2} \tag{13}$$

We conclude that for all $m > 2$ and any value of n, our assumption, $\overline{\mathcal{Z}}(n,m) \leq \frac{1}{2}$ is correct.□

Theorem 2: For all $m > 1$, $\overline{\mathcal{C}}(n,m) \leq \log(m) + 1$.

Proof: According to Eqs. 3 to 5 we have three possible cases to prove.

Case 1: $m \leq \alpha$ and $m \leq \beta$. $\log(m) + 1 = \log(2m)$, therefore, if we substitute $\overline{\mathcal{C}}(\alpha, i)$ by $\log(2i)$ in Eq. 3, we get

$$\overline{\mathcal{C}}(n,\, m) \leq \log(2) + \frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}} \log(2m) + \sum_{i=0}^{m-1} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \log(2(m-i)) \leq \log(2m) \tag{14}$$

The proof for **Case 2** ($m \leq \alpha$ and $m > \beta$) and **Case 3** ($m > \alpha$ and $m > \beta$) can be proven together starting with

$$\overline{\mathcal{C}}(n,\, m) = 1 + \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \overline{\mathcal{C}}\,(\alpha,\, m\, -\, i) \tag{15}$$

Using the fact that $\overline{\mathcal{C}}(\alpha, i) \leq \log(2i)$ yields,

$$\overline{\mathcal{C}}(n, m) \leq \log(2) + \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \log(2(m-i)) \leq \log(2m) \qquad (16)$$

We conclude that $\overline{\mathcal{Z}}(n, m) \leq \log(m) + 1$ is correct for any n and all $m > 1$.□

4 Throughput Analysis

Although CARMA-NTG does not require time slotting to operate, this section does assume slotting to simplify the analysis. The analysis in this section uses the same traffic model as for the slotted CARMA protocol presented in [6]. It is assumed that there is an infinite number of stations forming a Poisson source sending RTSs (both new and retransmitted) with an aggregate mean generation rate of λ RTSs per unit time. Each station is assumed to have at most one RTS to transmit at any given time. With this model, the average number of RTS arrivals in a time interval τ is $\lambda\tau$, i.e., $m = \lambda\tau$. All data packets have a duration of δ seconds, and the time to transmit a control packet is γ seconds. Both γ and δ are multiples of τ. The number of packets in a message is a random variable, and the probability that a message will complete its transmission (in a given cycle) is given by $q = \frac{1}{N}$ where N is the average number of packets in a message. We also make the same assumptions introduced in Section 3, and assume that the time to transition between transmit and receive states is negligible. To simplify our analysis, the members in the transmission group are ordered by the number of packets (in the message) remaining to be transmitted, and requests to be added to the transmission group is allowed even when the group is full. The later assumption makes our analysis a pessimistic lower bound on the performance of the protocol.

For a station to be added to the transmission group, the RTS/CTS exchange must be successful, the RTS must be the only one in the channel during its transmission. The probability of this happening is

$$P_s = P\{k = 1 \text{ arrival in a slot} | \text{some arrivals in a slot}\} = \frac{\lambda\tau \cdot e^{-\lambda\tau}}{1 - e^{-\lambda\tau}} \qquad (17)$$

The fraction of colliding RTSs is resolved with the help of the splitting algorithm, where one successful RTS/CTS exchange per contention period is guaranteed. Each successful RTS/CTS exchange has a duration of $T_s = 2\gamma + 2\tau$ seconds. Each collision has a duration $T_f = \gamma + \tau$ seconds, while each $\overline{\mathcal{Z}}$ has a duration of τ seconds. The upper bounds on average collision resolution costs are independent of the number of stations. The probability P_a that a station will be added to the transmission group is equal to 1, since per contention period the algorithm guarantees one successful RTS/CTS exchange. Each contention period attaches a new member to the transmission group.

The states of the group-transmission period can be represented by a Markov chain. Fig. 2 is an example for a network that allows at most $h = 4$ stations to be members of the group-transmission period. Transitions from one state to the other have been adequately labeled. The states in the Markov chain represent the probability P_k that k members are in the transmission group. Notice that the number of members in the transmission group can only increase by one, but it can decrease by up to k. At most, one new member can be added to the transmission group per contention period, but any number of members can be released from the group per group-transmission period. Transitions departing from and returning to the same state are not shown in Fig. 2, since what comes into the state is the same as what goes out of the state. They represent the case of one member being accepted into the transmission group, while another member leaves the transmission group.

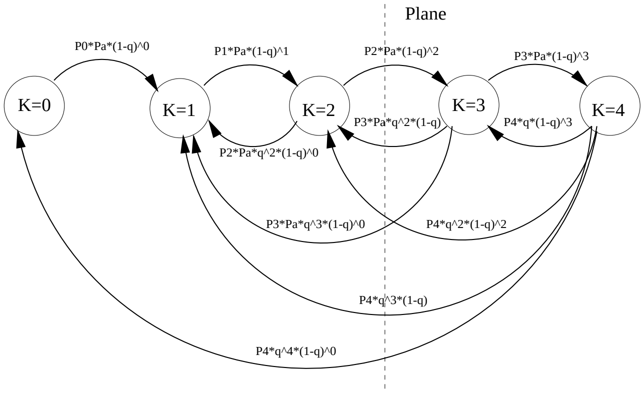

Figure 2: Markov Chain defining the transitions from one state to the other. The given example is for a network that allows, at the most, 4 members in the transmission group.

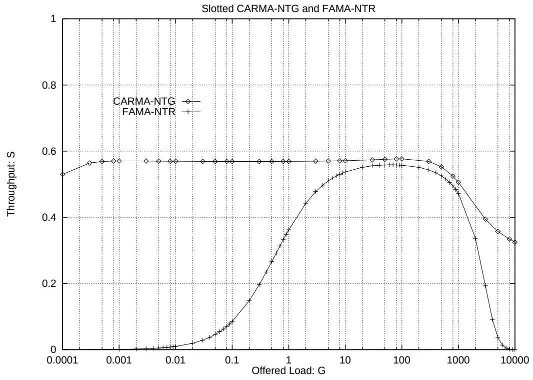

Figure 3: Approximate throughput comparison for slotted CARMA-NTG with an average number of group members $\rho = 1$ and for slotted FAMA-NTR with an average of one packet in the train $\rho = 1$. The comparison is done for low-speed network and a small data packet δ.

We can generalize our example and define h as the maximum number of members in the transmission group allowed by the network. If we draw a line between any two consecutive states of the Markov chain, then the flow of the transitions going in one direction has to be equal to the flow in the other direction. Therefore,

$$P_k P_a (1-q)^k = \sum_{i=1}^{h-k-1} P_a P_{k+i} \sum_{j=i+1}^{k+i} q^j (1-q)^{k+i-j} + P_h \sum_{j=h-k}^{h} q^j (1-q)^{h-j} \qquad (18)$$

If we divide both sides by $P_a(1-q)^k$, the result is

$$P_k = \sum_{i=1}^{h-k-1} P_{k+i} \left(\sum_{j=i+1}^{k+i} q^j (1-q)^{i-j} \right) + \frac{P_h}{P_a} \sum_{j=h-k}^{h} q^j (1-q)^{h-j-k} \qquad (19)$$

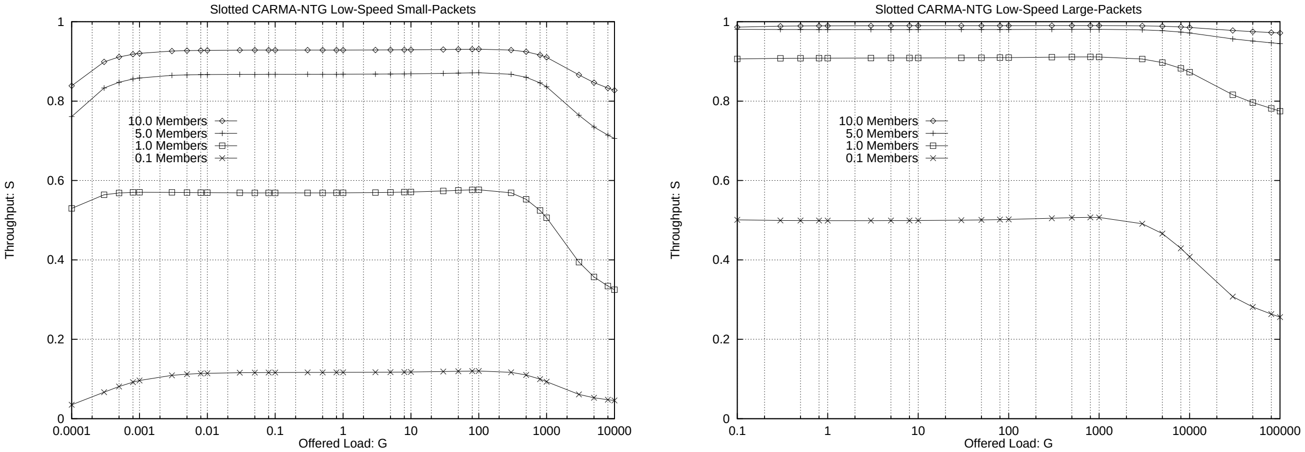

Figure 4: Approximate throughput comparison for slotted CARMA-NTG for low-speed networks when the average number of members in the transmission group ρ is allowed to vary. The comparison is done for both large and small data packets.

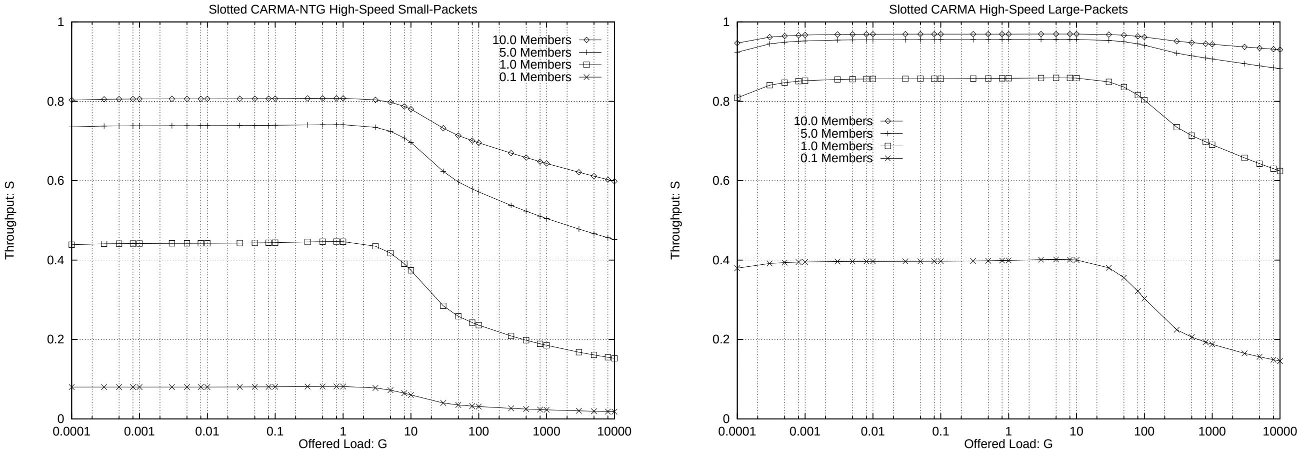

Figure 5: Approximate throughput comparison for slotted CARMA-NTG for high-speed network when the average number of members in the transmission group ρ is allowed to vary. The comparison is done for both large and small data packets.

which is valid for $0 \leq k \leq h-2$. Again $P_a = 1$. For $k = h-1$ the equation changes since the network cannot hold more than h members in the transmission group. There is no state to which to increase after P_h has been reached, in fact

$$P_{h-1} = \frac{P_h}{P_a} \sum_{j=1}^{h} q^j (1-q)^{1-j} \tag{20}$$

Eqs. 19 and 20 together with the fact that the sum of all probabilities $\sum_{i=0}^{h} P_i = 1$, form a system with $h+1$ equations with the same number of unknown variables. Starting with P_{h-1} we can make each of the equations be a function of P_h. To solve for P_h, we can substitute each of these terms in the equation $\sum_{i=0}^{h} P_i = 1$. Using P_h we can get a value for each of the remaining probabilities. Thus, the average number of group members is $\rho = \sum_{i=1}^{h} iP_i$. therefore, the average channel throughput is equal to

$$S = \frac{\overline{U}}{\overline{B_c} + \overline{B_g} + \overline{I}} \tag{21}$$

where $\overline{U}$ is the average utilization time of the channel, during which the channel is being used to transmit data packets; $\overline{B_c}$ is the expected duration of the contention period; $\overline{B_g}$ is the expected duration of the group-transmission period; and $\overline{I}$ is the average idle period, i.e., the average interval between two consecutive busy periods.

The average utilization $\overline{U}$ depends on the average number of members in the transmission group ρ and increases the throughput of the system.

$$\overline{U} = \rho\delta \tag{22}$$

The expected duration of the contention period $\overline{B_c}$ depends on the load of the channel. The contention period is not constant but varies based on the given load. The larger the contention period is, the lower the throughput is. If the load is high, the splitting algorithm requires more time to resolve the collision and have a successful RTS/CTS exchange. The duration of an average contention period equals the sum of the percentage of the successfully RTS/CTS exchange periods times their duration T_s, plus the percentage of the tree periods times their duration. The tree periods are composed of three parts each with a distinct cost and duration. According to the average cost results and the upper boundary for each of them, the average contention busy period can be written as

$$\begin{aligned}\overline{B_c} &= T_s \cdot P_s + \left[T_f \cdot (\log(\lambda\tau) + 1) + \frac{\tau}{2} + T_s\right](1 - P_s) \\ &= \frac{2(\gamma + \tau)\lambda\tau \cdot e^{-\lambda\tau}}{1 - e^{-\lambda\tau}} + \\ &\quad \frac{\left[(\gamma + \tau)(\log(\lambda\tau) + 3) + \frac{\tau}{2}\right](1 - e^{-\lambda\tau}(1 + \lambda\tau))}{1 - e^{-\lambda\tau}}\end{aligned} \tag{23}$$

The average group-transmission period is equal to the average number of group members (ρ) times the transmission period for one packet ($\delta + \tau$); therefore,

$$\overline{B_g} = \rho \cdot (\delta + \tau) \tag{24}$$

If the transmission group is empty, the length of the idle period is given by the next RTS arrival into the channel. In contrast, when the transmission group is not empty, the length of the idle period is limited by the start of the next transmission period. If no station is added to the transmission group, the idle period is limited to 2τ seconds; else, if one or more stations bid to be added to the transmission group, the idle period lasts τ seconds. Accordingly, we obtain

$$\begin{aligned}\overline{I} &= P_0\tau\frac{1}{1 - e^{-\lambda\tau}} + (1 - P_0)\left(\tau\frac{\lambda\tau e^{-\lambda\tau}}{1 - e^{-\lambda\tau}} + 2\tau(1 - \frac{\lambda\tau e^{-\lambda\tau}}{1 - e^{-\lambda\tau}})\right) \\ &= \frac{P_0\tau}{1 - e^{-\lambda\tau}} + (1 - P_0)\left(2\tau - \tau\frac{\lambda\tau e^{-\lambda\tau}}{1 - e^{-\lambda\tau}}\right)\end{aligned} \tag{25}$$

Substituting Eqs. 22, 23, 24, and 25 into Eq. 21 the average throughput can be written as:

$$S \geq= \frac{-2\rho\delta(1 - e^{-\lambda\tau})}{Ae^{-\lambda\tau} + B} \tag{26}$$

with

$$\begin{aligned}A &= (2\tau + 2\lambda\tau^2 + 2\gamma + 2\gamma\lambda\tau)\log(\lambda\tau) + \\ &\quad (5\lambda\tau^2 + 2\gamma\lambda\tau + 11\tau + 6\gamma + 2\rho\delta + 2\rho\tau - 2\lambda\tau^2 P_o - 4\tau P_o) \\ B &= [(-2\tau - 2\gamma)\log(\lambda\tau) + (-11\tau - 6\gamma - 2\rho\delta - 2\rho\tau + 2\tau P_o)]\end{aligned} \tag{27}$$

The average throughput for our algorithm can directly be compared with the results obtained in [5] for the slotted FAMA-NTR protocol. The throughput presented in [5] assumes that each station has at most one data packet to send. If we modify this assumption and allow an average packet train of size $\rho\delta$ the new throughput for slotted FAMA-NTR can be written as:

$$S_{fama} = \frac{\rho\delta\lambda\tau e^{-\lambda\tau}}{(\gamma\lambda\tau + \rho\delta\lambda\tau + \lambda\tau^2 - \gamma - 3\tau)e^{-\lambda\tau} + (4\tau + \gamma)} \tag{28}$$

The performance comparison is done for both low speed network (9600 bps) and high speed network (1 Mbps) with small data packets of 53 bytes (as in ATM cells) and longer data packets of 400 bytes. We assume the spacing between stations to be the same and define the diameter of the network to be 16.090 Km., which is 10 miles. Assuming these parameters, the propagation delay of the channel is 54μs. To accommodate the use of IP addresses for destination and source, the minimum size of RTSs and CTSs is 20 bytes. We normalize both throughput results by setting $\delta = 1$ and defining the following variables

$$\begin{aligned}a &= \frac{\tau}{\delta}\text{(normalized propagation delay)} \\ b &= \frac{\gamma}{\delta}\text{(normalized control packets)} \\ G &= \lambda \cdot \delta\text{(offered load, normalized to data packet)}\end{aligned} \tag{29}$$

If we substitute the new normalized variables from Eq. 29 into the throughput Eqs. 26 and 28, we obtain

$$S \geq \frac{-2\rho(1 - e^{-aG})}{A'e^{-aG} + B'} \tag{30}$$

with

$$\begin{aligned}A' &= (2a + 2a^2G + 2b + 2abG)\log(aG) + \\ &\quad (5a^2G + 2abG + 11a + 6b + 2\rho + 2a\rho - 2a^2GP_o - 4aP_o) \\ B' &= [(-2a - 2b)\log(aG) + (-11a - 6b - 2\rho - 2a\rho + 2aP_o)]\end{aligned} \tag{31}$$

for the throughput of our algorithm. For the slotted version of FAMA-NTR, the throughput can be written as:

$$S_{fama} = \frac{\rho aGe^{-aG}}{(a^2G + abG + \rho aG - 3a - b) \cdot e^{-aG} + (4a + b)} \tag{32}$$

Network Speed	Packet Size	δ	$a = \frac{\tau}{\delta}$	$b = \frac{\gamma}{\delta}$
9600 bps	424 bits	44166.7 μs	0.0012	0.377
9600 bps	3200 bits	333333.3 μs	0.000162	0.050
1 Mbps	424 bits	424 μs	0.127	0.377
1 Mbps	3200 bits	3200 μs	0.0168	0.050

Table 3: Protocol variables for low-speed networks (9600 bps) and high-speed networks (1 Mbps) with two types of data packets, small (424 bits) or large (3200 bits). The channel delay is $\tau = 54\mu$s, while the control packets are 160 bits long.

In Table 3, we summarize the protocol parameters. In Fig. 3 we compare the throughput for slotted CARMA-NTG to slotted FAMA-NTR for the case in which there is an average of one packet per packet train. The comparison is done for low-speed network 9600 bps and a small data packet $\delta = 53$ bytes. Figs. 4 and 5 show the throughput (S) versus the offered load (G) for varies average number of members in the transmission group ρ for slotted CARMA-NTG. The average number of group members ρ is a function of the average number of packets in a message N and can be calculated by solving the probabilities P_k in the Markov chain. As the average number of members in the transmission group increases due to increase in the offered load, the throughput of CARMA-NTG approaches the throughput of

TDMA; this would be the case if the stations engaged in long-term sessions. Eventually, the throughput of CARMA-NTR goes to 0 as in FAMA-NTR since the size of the contention increases with more stations (to get the RTS/CTS success requires more time) while the size of the transmission group is limited by the physical properties of the channel.

5 Conclusions

We have described and analyzed a specific protocol, CARMA-NTG, as an example of the integration of collision avoidance with dynamic transmission groups in a wireless network. CARMA-NTG dynamically divides the channel into cycles of variable length; each cycle consists of a contention period and a group-transmission period. During the contention period, a station with one or more packets to send competes for the right to be added to the transmission group; this is done using a collision-resolution-splitting algorithm based on a request-to-send/clear-to-send (RTS/CTS) exchange with carrier sensing.

Our analysis shows that CARMA-NTG provides high throughput when either a small or a large number of stations need to access the channel. Allowing the maximum size of the transmission group to equal the number of stations in the system, CARMA-NTG becomes TDMA in effect when all stations need to transmit. CARMA-NTG is much more efficient than TDMA when there are only a few stations with packets to send .

Our design and analysis is limited to fully connected wireless networks using a single channel. Our work continues to address more detailed analysis of performance of these types MAC protocols [13], more sophisticated group allocation strategies, and the application of these type of protocols to networks with multiple hops and multiple channels.

References

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: a media access protocol for wireless LANs,” *Proceedings of ACM SIGCOMM ’94*, pp. 212–25, ACM, 1994.

[2] K. Biba, “A Hybrid Wireless MAC Protocol Supporting Asynchronous and Synchronous MSDU Delivery Services,” *Tech. Rep. Paper 802.11/91-92*, IEEE 802.11 Working Group, 1992.

[3] R. L. Brewster and A. M. Glass, “Throughput analysis of non-persistent and slotted non-persistent CSMA/CA protocols,” *4th International Conference on Land Mobile Radio*, pp. 231–6, Institution of Electronic and Radio Engineers, 1987.

[4] C. Fullmer and J.J. Garcia-Luna-Aceves, “FAMA-PJ: a channel access protocol for wireless LANs,” *Proc. ACM Mobile Computing and Networking ’95*, Berkeley, CA, Nov. 14-15, 1995.

[5] C. Fullmer and J.J. Garcia-Luna-Aceves, “Floor acquisition multiple access for packet-radio networks,” *Proc. ACM SIGCOMM ’95*, Cambridge, MA, August 30-September 1, 1995.

[6] R. Garcés and J.J. Garcia-Luna-Aceves, “Floor acquisition multiple access with collision resolution,” *Proc. ACM/IEEE Mobile Computing and Networking ’96*, Rye, NY, Nov. 10-12, 1996.

[7] P. Karn, “MACA - a new channel access method for packet radio,” *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–40, ARRL, 1990.

[8] M.J. Karol and I. Chih-Lin, “A protocol for fast resource assignment in wireless PCS,” *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 727-32, IEEE, 1994.

[9] L. Kleinrock and M. Scholl, “Packet switching in radio channels: new conflict-free multiple access schemes,” *IEEE Trans. Commun.*, vol. COM-28, no. 7, pp. 1015–1029, 1980.

[10] L. Kleinrock and F. A. Tobagi, “Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics,” *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, 1975.

[11] W. F. Lo and H. T. Mouftah, “Carrier sense multiple access with collision detection for radio channels,” in *IEEE 13th International Communications and Energy Conference*, pp. 244–47, IEEE, 1984.

[12] A. Muir and J.J. Garcia-Luna-Aceves, “Group allocation multiple access in single-channel wireless LANs,” *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference*, Phoenix, AZ, 1997.

[13] A. Muir and J.J. Garcia-Luna-Aceves, “group allocation multiple access with collision detection,” *Proc. IEEE INFOCOM 97*, Kobe, Japan, April 7–11, 1997.

[14] G. S. Sidhu, R. F. Andrews, and A. B. Oppenheimer, *Inside AppleTalk, Second Edition*. Addison-Wesley Publishing Company, Inc., 1990.

[15] A. S. Tannenbaum, “Computer Networks,” *Prentice Hall, Inc.*, New Jersey, 1981.

[16] L. G. Roberts, “Dynamic allocation of satellite capacity through packet,” *Computer Communication Network*, Netherlands, 1975.

[17] I. Rubin, “Access control disciplines for multi-access communications channels: reservation and TDMA schemes,” *IEEE Trans. on Information Theory*, vol. IT-25, no. 5, pp. 516–536, 1979.